Real-time Cohorting of Nursing Care into Bubbles

Jeffrey Keithley
University of Iowa
Iowa City, USA
jeffrey-keithley@uiowa.edu

Tinh Tran
University of Iowa
Iowa City, USA
integratingjava@gmail.com

Lucas Zach-Ryan
University of Iowa
Iowa City, USA
lucas.j.zach@gmail.com

D. M. Hasibul Hasan
University of Iowa
Iowa City, USA
simoncuet@gmail.com

Brodie McCuen
University of Iowa
Iowa City, USA
mccuen.brodie@gmail.com

Sriram V. Pemmaraju
University of Iowa
Iowa City, USA
sriram-pemmaraju@uiowa.edu

Bijaya Adhikari
University of Iowa
Iowa City, USA
bijaya-adhikari@uiowa.edu

ABSTRACT

Healthcare facilities, such as hospitals and long-term care facilities, which house vulnerable populations, are sites for pathogens (such as Antibiotic-Resistant Organisms, *Clostridioides difficile*, influenza viruses, and SARS-CoV-2, among others) to spread. The contacts between healthcare providers (HCPs) and the patients that naturally emerge during the course of care delivery also serve as pathways for the infections to flow. Prior work has identified that cohorting patients and HCPs into nearly isolated groups leads to reduction in infection. However, most of these works are either too simplistic (e.g. cohorting after observing infections) or have limited practical use (e.g. retrospective cohorting). In this paper, our goal is to minimize infection spread in real-time by creating cohorts of HCPs and patients on-the-fly as patients are being admitted, discharged or transferred.

Specifically, we formulate the novel Online Bubble Clustering Problem, which asks to create and maintain cohorts of HCPs and patients that have limited external contacts, yet meet the care demands of the patients and care capacity of the HCPs. We also theoretically demonstrate that the problem is very challenging in both deterministic and stochastic settings, implying no algorithm could achieve results close to the optimal solution. Despite the hardness, we propose natural heuristics and design offline an integer linear programming (ILP) approach to contrast the heuristics against the optimal solution. We also conduct extensive agent-based modeling experiments on granular HCP mobility data collected using sensor systems we previously deployed in a medical intensive care unit over a 30-day period, overlaid with a COVID-19 agent-based disease model. Our simulation results show that real-time cohorting leads to significantly lower disease prevalence, cutting cases in half in some scenarios, while maintaining reasonably low levels of excess mobility and HCP workload.

KEYWORDS

online patient cohorting, healthcare-associated infections, greedy algorithms, integer linear programming

INTRODUCTION

The amplification of COVID-19 in healthcare facilities has clearly exposed the vulnerability of our healthcare systems to nosocomial outbreaks [8]. Similarly, the burden of nosocomial spread of other infections such as influenza and infections caused by pathogens such as *Clostridiodes Difficle*, *Vancomycin Resistant Enterococcus*, *Methicillin-resistant Staphylococcus* aureus are well documented [6, 28]. These *healthcare-associated infections (HAIs),* cost an estimated $28-45 billion in the US alone [29], also leading to more than 2.5 million (estimated) new cases and 33,310 deaths per year in Europe [7]. Hence, designing prevention strategies for infection spread within healthcare facilities is paramount.

Given its importance, intervention against HAIs has garnered significant research interest. Studies have focused on hand hygiene [20, 26], isolating patients with HAIs to prevent further transmission [26], disinfecting surfaces to reduce environmental contamination [2, 10], antibiotic stewardship [24], usage of personal protective equipment [22], and improved surveillance [17, 18]. Prior studies have also examined *cohorting* as a means of infection control [1, 10].

Cohorting practices typically confine patients with the same infection to a specific region within a healthcare facility to prevent further transmission. Prior work on cohorting focuses on grouping symptomatic patients [16] and potentially assigning specific healthcare professionals (HCPs) to these patients [1]. These strategies have three critical drawbacks; *i)* cohorting takes place only after many patients show symptoms, which could be too late, *ii)* cohorting is oblivious to asymptomatic infections, which would lead to asymptomatic (but colonized) individuals being grouped with susceptible individuals, and *iii)* HCP cohorting could be challenging to implement because certain HCPs may have specialized skills that are necessary for patients in all cohorts.

A different approach to cohorting was considered in [11], where patients and HCPs are preemptively clustered into "bubbles", which refer to groups of individuals primarily interacting with each other, while limiting outside interaction. A critical difference between this approach and standard cohorting practices is that bubble clustering groups individuals based on their healthcare needs in order to minimize external interactions; this grouping is done preemptively and

not in response to emerging symptoms. As the clustering minimizes external interactions of a group, any infection (including asymptomatic) originating in a bubble is likely to be contained within. While [11] demonstrated the usefulness of bubble clustering, their result is essentially existential. Their algorithm for computing a bubble clustering was retrospective, i.e., it assumed full knowledge of the events that have already occurred in a hospital unit (e.g., admissions, discharges, patient characteristics, HCP mobility, etc.).

Practicality and Contributions. To be practical and implementable in a healthcare facility, a bubble clustering approach must maintain and update the bubbles in real-time as patients are admitted and discharged with minimal knowledge about future events. This implies that the resulting computational problem is "online" [31] in nature (See section "Problem Formulation and Hardness"). To this end, in consultation with an infectious disease clinician and ICU nursing staff at a large tertiary hospital, we formulate a novel Online Bubble Clustering (OBC) problem which seeks to maintain bubbles of patients (and their rooms) and HCPs in real-time while minimizing interactions across bubbles. Note that we view bubble clustering as a preemptive infection control measure that is expected to be integrated into routine hospital operations and therefore it is critical that bubble clustering not lead to any adverse impacts on patient care. Keeping this in mind, the OBC problem contains constraints that aim to bound HCP workload (so the HCPs are not overworked) and bubble diameter (so the rooms in each bubble are nearby) constraints. We then theoretically establish OBC's hardness, showing that any algorithm (whether deterministic or stochastic) that does not have knowledge of future events, is bound to perform extremely poorly *in the worst case* in comparison to the optimal offline solution. While the worst-case hardness of the OBC problem is discouraging, real-world data is often structurally quite different from worst case instances. Thus, we propose three simple, easy-to-explain online algorithms for the OBC problem. We also model the offline version of the bubble clustering problem as an integer linear program in order to compute an optimal offline solution and empirically benchmark our online heuristics. We conduct extensive agent-based modeling experiments using real-world (anonymous) patient and hospital operations data collected from a Medical Intensive Care Unit (MICU) in a large tertiary care healthcare facility. Our results demonstrate that bubble clustering is indeed feasible and effective in real-time, i.e., there are simple online algorithms that are able to maintain bubble clusters which satisfy HCP workload and mobility constraints, while leading to effective infection control.

PROBLEM FORMULATION AND HARDNESS

Consider the decisions made by a hospital nurse manager in an Intensive Care Unit (ICU) who is responsible for processing patients as they are admitted into and discharged from the ICU. The nurse manager is aware of each arriving patient's medical history (e.g., diagnoses, test results, procedures, medications, etc.) and therefore has an estimate of the patient's need for care. They then admit the patient to an available room in the ICU given this information. We view this decision as an opportunity to cohort patients and HCPs into bubbles as a preemptive infection control measure. Prior work analyzed HCP mobility data from an ICU to show that cohorting naturally emerges to some extent, possibly due to choices made by individual HCPs [23]; here we aim to engineer cohorting via the decisions made by the nurse manager. However, these decisions have to be made while ensuring that there is no degradation in patient care nor an inordinate increase in each HCP's workload. We call this (informally stated problem) the *Online Bubble Clustering* problem and formulate it precisely below. We illustrate a toy instance of the bubble clustering problem and the corresponding clustering solution in Figure 1. The formal description of the problem requires some background on online allocation problems, which we present next, followed by a description of how we model different elements of the problem.

Background on Online Allocation Problems: An *online allocation* problem is presented a sequence of input items that are revealed one at a time, and algorithm Alg seeks to make an irrevocable decision in response to each item's arrival to optimize an objective $f(\cdot)$ while satisfying some predefined constraints. Alg is said to have *competitive ratio* of ρ for a minimization problem if we have $f(\text{Alg}) \leq \rho \cdot f(\text{OPT})$, for all sufficiently large inputs. Here, $f(\text{OPT})$ is the optimal value of $f(\cdot)$ computed using full knowledge of all items. For example, the input to the classic Online Bin Packing problem consists of a sequence of items with size in [0, 1], revealing them one at a time. An algorithm has to irrevocably pack the arriving item immediately into a bin with max capacity of 1, with the goal of minimizing the total number of bins used. A well-known result for Online Bin Packing is that a simple First Fit algorithm, which places an item into the first bin the item fits into, has a competitive ratio of 1.7 [21]. The online setting has been used for a variety of problems, e.g., clustering, matching, hiring, submodular optimization, paging, job scheduling, etc. See [31] and the references therein.

Patient Events. The input consists of a sequence of patient events $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_T)$ for timesteps $0 < t_1 \leq t_2 \leq \cdots \leq t_T$, where each event $\mathcal{E}_i$ occurs at time t_i and has type admission or discharge, which we denote by $type(\mathcal{E}_i)$. Each event $\mathcal{E}_i$ corresponds to a patient visit v. Let $\mathcal{V}$ denote the set of patient visits that the events in $\mathcal{E}$ correspond to. Conversely, each patient visit $v \in \mathcal{V}$ is associated with exactly two events, $\mathcal{E}_a$ and $\mathcal{E}_b$, where $t_a \leq t_b$ and $type(\mathcal{E}_a) = \text{admission}$ and $type(\mathcal{E}_b) = \text{discharge}$. In this case, we say that the *visit interval* I_v associated with visit v is $I_v = (t_a, t_b)$ and the *length of stay* associated with visit v (denoted LOS_v) is $\text{LOS}_v = \max(1, t_b - t_a)$.

HCPs. We are given a fixed set P of HCPs, which is partitioned into a set P_s of *substitutable* HCPs and a set P_{ns} of *non-substitutable* HCPs. An example of a substitutable HCP is a nurse; typically, a unit such as an ICU will consist of several nurses who can substitute for each other. A non-substitutable HCP is a specialist such as a cardiologist or a pulmonologist, who visits the unit once or twice during the day and whose expertise may not be found in any other HCP associated with the unit. The set P_s of substitutable HCPs is further partitioned into $P_{s,day}$ and $P_{s,night}$, representing the substitutable HCPs present during the day and night shifts, respectively. Members of $P_{s,day}$ can substitute for each other and similarly members of $P_{s,night}$ can substitute for each other. We associate a *load* L_p with each HCP $p \in P$, which is defined as the average amount of time (in minutes) the HCP p spends with patients each day.

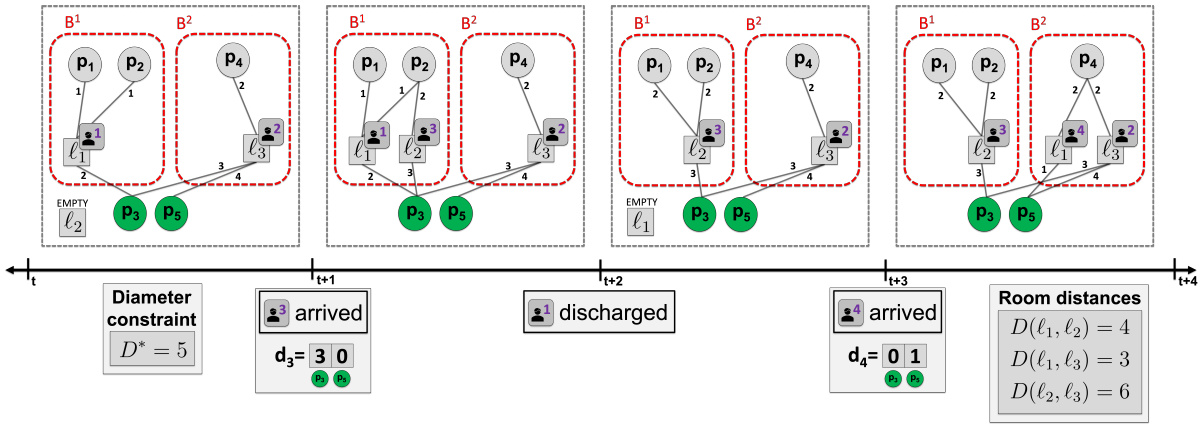

Figure 1: An example of a bubble clustering as patients arrive in a small hospital unit illustrating three events, at times $t+1, t+2, t+3$. Substitutable HCPs $P_1 = \{p_1, p_2, p_4\}$ are grey and non-substitutable $P_2 = \{p_3, p_5\}$ are green. There are three locations $\mathcal{R} = \{\ell_1, \ell_2, \ell_3\}$ and 4 patient visits to the hospital unit. Initially, i.e., prior to time $t+1$, patient visits 1 and 2 have already been assigned to rooms and bubbles. Substitutable HCPs and patients are assigned to one of two bubbles - note that unoccupied rooms (labeled "empty") are not assigned to any bubble. As each patient arrives, they get assigned to an empty room, which is then assigned to a bubble according to some policy (room/bubble assignment policies are discussed in the "Algorithmic Approach" section). Placing patient visit 3 in bubble 2 violates the diameter constraint, so this visit must be placed in bubble 1. Placement in either bubble is feasible for patient visit 4, and choosing bubble 2 incurs a lower cross-bubble demand (patient demands are shown as d_3 and d_4 and as edge labels).

Metric Space on Locations. Let $\mathcal{R}$ denote a fixed set of locations (e.g., patient rooms) that patients are admitted to upon arrival. We assume access to a function $D : \mathcal{R} \times \mathcal{R} \to \mathbb{R}_+$ that provides distances between each pair of locations. We interpret $D(\ell_1, \ell_2)$ to be the walking distance between room ℓ_1 and room ℓ_2.

Online Processing of Patient Events. For a fixed positive integer K, let $B^1, B^2, \ldots, B^K$ denote the *bubbles* we wish to maintain as patient events occur. As part of the initialization, the set of substitutable day shift HCPs $P_{s,day}$ and the set of substitutable night shift HCPs $P_{s,night}$ are partitioned in some manner into the K bubbles (e.g., equitably distributing HCP loads among the bubbles). For the scope of this paper, we evenly distribute substitutable HCPs among bubbles, but this assumption could be relaxed. The patient events are then processed in the order in which they occur. Algorithm 1 shows the *irrevocable* decisions made during the processing of the patient event $\mathcal{E}_i$. We suppose that v is the patient visit associated with event $\mathcal{E}_i$ and if $\mathcal{E}_i$ is a discharge event, we assume that (ℓ, v), for some $\ell \in \mathcal{R}$ has been previously added to bubble B^k for some $k \in [K]$.

The crux of the problem we are faced with is how to pick an available location $\ell \in \mathcal{R}$ and a bubble B^k, $k \in [K]$ (Lines 2-3), in response to an admission event. Note that these two decisions are made without any knowledge of when visit v will end or any other future events occurring after time t_i.

Patient Demand Vectors. The estimated care required by a patient during visit v is encoded as a $2 + |P_{ns}|$ dimensional *patient demand* vector $\mathbf{d}_v \in \mathbb{R}_+^{\{day,night\} \cup P_{ns}}$, where each element represents estimated care needed in minutes per day. The first two elements of $\mathbf{d}_v$ correspond to the estimated care required during visit v from day and night shift substitutable HCPs, respectively (denoted $\mathbf{d}_v^{\mathbf{s}}$). The remaining $|P_{ns}|$ elements of $\mathbf{d}_v$ correspond to the estimated care required from each non-substitutable HCP in P (denoted $\mathbf{d}_v^{\mathbf{ns}}$). Note that $\mathbf{d}_v$ is therefore the concatenation $\mathbf{d}_v^{\mathbf{s}} \cdot \mathbf{d}_v^{\mathbf{ns}}$, and $\mathbf{d}_v$ is only revealed at the admission event that starts visit v.

Algorithm 1 Online Processing of Patient Events

1: **if** $type(\mathcal{E}_i)$ = admission **then**
2: visit v is assigned to available location $\ell \in \mathcal{R}$
3: (ℓ, v) is added to a bubble B^k, $k \in [K]$
4: ℓ is designated unavailable
5: **end if**
6: **if** $type(\mathcal{E}_i)$ = discharge **then**
7: (ℓ, v) is removed from bubble B^k
8: ℓ is designated available
9: **end if**

Objective Function: Cross-Bubble Demand. The purpose of bubble clustering is to contain disease-spread within the bubbles. However, the bubbles are not "air tight" because patients can have contact with non-substitutable HCPs who are outside all bubbles. Thus, the metric we seek to minimize, which we call *cross-bubble demand*, is the total amount of time each non-substitutable HCP spends with patients in distinct bubbles. We view this time as an opportunity for the non-substitutable HCP to contract infection or pick up a pathogen from a patient in one bubble and spread it to a patient in a different bubble. To make this precise, we introduce some notation. Let $\mathcal{B}_i = (\mathcal{B}_i^1, \mathcal{B}_i^2, \ldots, \mathcal{B}_i^K)$ denote the set of the bubbles immediately after event $\mathcal{E}_i$ has been processed. Let $T_i(I_v, I_{v'}) = 1$ if I_v and $I_{v'}$ overlap at time t_i (and 0 otherwise) and let $T(I_v, I_{v'}) = \sum_{i=1}^{T} T_i(I_v, I_{v'})$. Let $B(v)$ denote the bubble which v is assigned to and let $s(B)$ denote the substitutable HCPs assigned to bubble B. The total-cross bubble demand is then defined as:

$$f((\mathcal{B}_i)_{i=1}^T) = \sum_{\substack{v,v' \in \mathcal{V} \\ B(v) \neq B(v')}} (\mathbf{d}_v^{\mathbf{ns}} \cdot \mathbf{d}_{v'}^{\mathbf{ns}}) \cdot T(I_v, I_{v'}) \quad (1)$$

Diameter and Load Constraints. While cohorting patient care into bubbles can limit disease-spread, it could have detrimental downstream effects on patient care quality. For example, HCPs could end up being overburdened relative to the setting with no cohorting. We focus on two costs to HCPs. First, the *diameter* of a bubble is the maximum distance between any two rooms in the same bubble, which measures how far individual HCPs must walk in order to care for patients in their bubble. Formally, the maximum diameter of the bubble clustering over all time t_i, $1 \leq i \leq T$, is

$$\hat{D}((\mathcal{B}_i)_{i=1}^T) = \max_{i=1}^{T} \max_{B \in \mathcal{B}_i} \max_{\ell,\ell' \in B} D(\ell, \ell') \quad (2)$$

Second, the total excess load is defined as

$$\hat{L}((\mathcal{B}_i)_{i=1}^T) = \sum_{i=1}^{T} \sum_{B \in \mathcal{B}_i} \left(\sum_{v \in B} \|\mathbf{d}_v^{\mathbf{s}}\|_1 - \sum_{p \in s(B)} L_p \right) \quad (3)$$

With these elements in place, the *Online Bubble Clustering (OBC)* problem can be stated as follows.

Online Bubble Clustering (OBC)
Given a patient event sequence $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_T)$, set of HCPs $P = (P_{s,day}, P_{s,night}, P_{ns})$, HCP loads $L : P \to \mathbb{R}_+$, a metric space $(\mathcal{R}, D)$ of locations in the unit, a patient demand vector $\mathbf{d}_v$ for each patient visit $v \in \mathcal{V}$, number of bubbles $K \in \mathbb{Z}_+$, and diameter and excess load upper bounds $D^*, L^* \in \mathbb{R}_+$, perform an online processing of patient events (see Algorithm 1), such that the cross-bubble demand $f((\mathcal{B}_i)_{i=1}^T)$ is minimized while satisfying the constraints $\hat{D} \leq D^*$ and $\hat{L} \leq L^*$.

We refer to a fully specified instance of the OBC problem as $\mathcal{M} = (\mathcal{E}, P, L, (\mathcal{R}, D), (\mathbf{d}_v)_{v \in \mathcal{V}}, K, B^*, L^*)$.

Hardness of Online Bubble Clustering

In this section, we establish that OBC is extremely hard to solve, even by *randomized* online algorithms. Our results show the power of foresight from the lens of worst case analysis. In other words, from a worst case perspective, lack of knowledge of future events can be extremely costly to any algorithm for OBC. We start with a hardness result for deterministic online algorithms (Theorem 1), which we then extend to randomized online algorithms (Theorem 2) using Yao's minimax principle [34].

Theorem 1. *For any deterministic online algorithm $\mathcal{A}$, there exists an input patient event sequence with T patient visits such that the cross-bubble demand for algorithm $\mathcal{A}$, $f((\mathcal{B}_i^{\mathcal{A}})_{i=1}^T) = \Omega(T \cdot d^2)$, whereas the cross-bubble demand for OPT, $f((\mathcal{B}_i^{OPT})_{i=1}^T) = 0$. Here d is the maximum patient demand for a non-substitutable HCP in the input instance.*

Proof: Let $n \geq 3$ be an odd integer and without loss of generality let T be a multiple of $2n$. Suppose there is a partition of $\mathcal{R}$ into subsets $\mathcal{R}_1$, $|\mathcal{R}_1| = \frac{n-1}{2}$ and $\mathcal{R}_2$, $|\mathcal{R}_2| = \frac{n-1}{2} + 1$, such that the diameter of $\mathcal{R}_1$ and the diameter of $\mathcal{R}_2$ are both at most D^*. Further, suppose that the rooms in $\mathcal{R}_1$ are far apart from the rooms in $\mathcal{R}_2$, i.e., for any $r_1 \in \mathcal{R}_1$ and $r_2 \in \mathcal{R}_2$, $D(r_1, r_2) > D^*$. Finally, suppose that K, the number of bubbles, is set to 2.

First, we claim that for any deterministic online algorithm $\mathcal{A}$ there is a patient event sequence with care demands such that $\mathcal{A}$ will incur a cross-bubble demand of at least $\Omega(d^2 \cdot T)$, where d is the maximum patient demand for a non-substitutable HCP in the input instance. Suppose the patient event sequence consists of $T/2n$ batches, where each batch consists of n patient admissions followed by n patient discharge events. The patient admissions within each batch are further partitioned into two groups.

- **Group 1**: admit $\frac{n-1}{2}$ patients with demand vector $d \cdot \mathbf{e}_1$ and then
- **Group 2**: admit $\frac{n-1}{2}$ patients with demand vector $d \cdot \mathbf{e}_2$.

After algorithm $\mathcal{A}$ has processed this patient event sequence, there is one available room in $\mathcal{R}$, which we call r. Due to distance constraints, all occupied rooms in $\mathcal{R}_1$ are in one bubble, say B^1, and all occupied rooms in $\mathcal{R}_2$ are in B^2.

If $r \in \mathcal{R}_1$, then algorithm $\mathcal{A}$ has no choice but to place the next arriving patient in room r and assign the room to B^1. Knowing this, the adversary admits a patient with demand vector $d \cdot \mathbf{e}_1$, if there are more Group 1 patients in B^2; otherwise the adversary admits a patient with demand vector $d \cdot \mathbf{e}_2$. In either case, the admission of the last patient in a batch causes algorithm $\mathcal{A}$ to incur a cross-bubble demand of at least $d^2 \cdot \frac{n-1}{4}$. The argument is symmetric if $r \in \mathcal{R}_2$. Summed over all $T/2n$ batches, the algorithm incurs a cross-bubble demand of at least $d^2 \cdot \frac{n-1}{4} \cdot \frac{T}{2n} = \Omega(d^2 \cdot T)$. An optimal offline algorithm will foresee the type of the last patient (i.e., Group 1 type or Group 2 type) in a batch, and place all $\frac{n-1}{2}$ patients of the smaller group in the batch in the $\mathcal{R}_1$ rooms and assign these to B^1 and place all $\frac{n-1}{2} + 1$ patients of the larger group in the $\mathcal{R}_2$ rooms and assign these to B^2. This incurs a cross-bubble demand of 0 per batch and therefore for all the batches as well. □

Somewhat surprisingly, we show that the above result for deterministic algorithms holds even for randomized algorithms, assuming the standard *oblivious adversary* setting, in which the adversary knows the "source code" of the online algorithm, but not the sequence of random bits used by the algorithm. This result shows that algorithmic randomness is not enough to counter the lack of foresight that is fundamental to the online setting. The proof of the following theorem builds upon the proof of Theorem 1 by applying Yao's minimax principle [34], and appears in the Technical Appendix.

Theorem 2. *For any randomized online algorithm $\mathcal{A}$, assuming an oblivious adversary, there exists an input patient event sequence with T patient visits such that the cross-bubble demand for algorithm $\mathcal{A}$, $\mathbb{E}[f((\mathcal{B}_i^{\mathcal{A}})_{i=1}^T)] = \Omega(T \cdot d^2)$, whereas the cross-bubble demand for OPT, $f((\mathcal{B}_i^{OPT})_{i=1}^T) = 0$. Here d is the maximum patient demand for a non-substitutable HCP in the input instance.*

Proof: Let $\mathcal{I}$ denote the set of all possible input sequences considered in the proof of Theorem 1 (see main paper). Recall that for each batch, the adversary considers two possible choices for the last patient admission event in that batch: admit a patient with demand $d \cdot \mathbf{e}_1$ or admit a patient with demand $d \cdot \mathbf{e}_2$. We now define a probability distribution on $\mathcal{I}$ by making each of these two choices

independently, with equal probability. Note that this induces a uniform distribution over the $2^{T/2n}$ elements in $\mathcal{I}$.

We now use Yao's Principle [34], which states that the expected cost of the best randomized online algorithm on the worst-case input in the oblivious adversary setting is at least the expected cost of the best deterministic online algorithm against a random worst-case input distribution. (Yao's principle is more general and applies to non-online algorithms also.)

As argued in the proof of Theorem 1, the deterministic algorithm $\mathcal{A}$ incurs a cost of at least $d^2 \cdot \frac{n-1}{4}$ for a batch, on at least one of the two inputs, implying that the expected cross-bubble demand of $\mathcal{A}$ is at least $d^2 \cdot \frac{n-1}{8}$ for a batch. By linearity of expectation, the expected cross-bubble demand of $\mathcal{A}$ is at least $d^2 \cdot \frac{T}{32}$ over all $T/2n$ batches. Then by Yao's Principle [34], this is a lower bound on the cost of any randomized algorithm for the OBC problem. □

We emphasize that these hardness results are quite strong. Since the offline optimal cross-bubble demand is 0, the competitive ratio for both the deterministic and randomized algorithms is infinite. Our results show that even the *regret*, i.e., the difference $\mathbb{E}[f((\mathcal{B}_i^{\mathcal{A}})_{i=1}^T)] - f((\mathcal{B}_i^{OPT})_{i=1}^T)$, grows linearly in the number of events and exponentially in the number of bits needed to represent the patient demands. Our results are also robust in the sense that if we allow the algorithm (but not the optimal offline solution) to use $O(K)$ bubbles (rather than exactly K bubbles), or bubbles with arbitrarily large diameter, the lower bounds still hold asymptotically (as shown in Theorem 3, whose proof is included in the Technical Appendix).

Theorem 3. *For any randomized online algorithm $\mathcal{A}$, assuming an oblivious adversary, there exists an input patient event sequence with T patient visits such that the cross-bubble demand for algorithm $\mathcal{A}$, $\mathbb{E}[f((\mathcal{B}_i^{\mathcal{A}})_{i=1}^T)] = \Omega(T \cdot d^2)$, even when it uses arbitrarily many bubbles, whereas the cross-bubble demand for OPT, $f((\mathcal{B}_i^{OPT})_{i=1}^T) = 0$, even when it uses just 2 bubbles. Here d is the maximum patient demand for a non-substitutable HCP in the input instance.*

Proof: Suppose that $K \geq 2$ and let $\mathcal{A}$ be a deterministic algorithm that is allowed to use K bubbles. Consider the event sequence described in the proof of Theorem 1 and consider the situation after $\mathcal{A}$ has processed the event sequence of $\frac{n-1}{2}$ Group 1 patient admissions and $\frac{n-1}{2}$ Group 2 patient admissions. As in the proof of Theorem 1, let $r \in \mathcal{R}$ denote the only available room at this stage and suppose that $r \in \mathcal{R}_1$. This means that the next arriving patient has to be admitted into r and placed in a bubble that does not contain any rooms in $\mathcal{R}_2$ (because that would violate the diameter constraint). Let us call a bubble containing a room in $\mathcal{R}_2$ an *inadmissible* bubble. Note that the inadmissible bubbles in total contain at least $\frac{n-1}{2}$ rooms. The adversary now admits a patient with demand $d \cdot \mathbf{e}_1$ if there are more Group 1 patients in inadmissible bubbles. Otherwise, the adversary admits a patient with demand $d \cdot \mathbf{e}_2$. In either case, the admission of the last patient in a batch causes algorithm $\mathcal{A}$ to incur a cross-bubble demand of at least $d \cdot \frac{n-1}{4}$. The argument is symmetric if $r \in \mathcal{R}_2$. Summed over all $T/2n$ batches, the algorithm incurs a cross-bubble demand of at least $d^2 \cdot \frac{n-1}{4} \cdot \frac{T}{2n} = \Omega(d^2 \cdot T)$.

As in the proof of Theorem 1, an optimal offline algorithm can process the event sequence, using just 2 bubbles, while incurring a cross-bubble demand of just 0.

It does not help the algorithm to increase the diameter of bubbles because the same argument as above will apply by simply separating the rooms in $\mathcal{R}_1$ and $\mathcal{R}_2$ further apart.

Finally, the application of Yao's minimax lemma (exactly as in the proof of Theorem 2 above) yields the hardness result for randomized algorithms. □

This is in contrast with results for online clustering problems (e.g., online k-means [4]) for which allowing extra clusters is critical to circumvent hardness results.

ALGORITHMIC APPROACH

Here we present algorithmic approaches that might be used by a nurse manager to assign each incoming patient visit v to a location $\ell \in \mathcal{R}$ and bubble $k \in [K]$. Recall that if the current event $\mathcal{E}_i$ is a discharge, then there is no decision to be made and the outgoing patient is simply removed from their location and bubble. Now suppose that $\mathcal{E}_i$ is an admission event with v being the associated patient visit. We assume that there is a user-specified subroutine called PairSelection that is called to process admission events.

For the scope of this paper, we consider the following three simple, transparent, and easy-to-explain instantiations of PairSelection for selecting a location-bubble pair to assign the incoming patient. In order to explain these instantiations, we introduce some notation. We define $f((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$ to be the cross-bubble demand function f evaluated over i sets of bubbles, with the first $i-1$ sets of bubbles given by $(\mathcal{B}_j)_{j=1}^{i-1}$ and the bubble set $\mathcal{B}_i$ obtained from $\mathcal{B}_{i-1}$ by adding visit-room pair (v, k) to bubble B_{i-1}^k. The functions $\hat{D}((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$ and $\hat{L}((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$ are defined similarly.

- **Random** selects, uniformly at random, a location-bubble pair from the set of *feasible* pairs

$$\mathcal{F} := \{(\ell, k) \mid \hat{D}((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k) \leq D^*, \hat{L}((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k) \leq L^*\}.$$

- **Greedy** selects a location-bubble pair from $\mathcal{F}$ that (locally) minimizes the total cross-bubble demand (Eq (1)) up to the current event, i.e., $\operatorname{argmin}_{(\ell,k)\in\mathcal{F}} f((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$.
- **τ-Greedy** can be viewed as a version of the greedy algorithm that attempts to balance the goal of minimizing current cross-bubble demand with the goal of keeping bubble diameter and excess load small for future arrivals. Specifically, the algorithm obtains a set $\mathcal{F}'$ by refining the feasible set $\mathcal{F}$ by including only those location-bubble pairs whose cross-bubble demand is at most $(1+\tau)$ times the minimum cross-bubble demand in $\mathcal{F}$, for a parameter $\tau > 0$. τ-Greedy then selects a location-bubble pair from $\mathcal{F}'$ that minimizes a linear combination of bubble diameter and excess load defined by $C((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k) = \alpha \cdot \hat{D}((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k) + (1-\alpha) \cdot \hat{L}((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$, where $\alpha \in [0, 1]$. That is, τ-Greedy chooses the location-bubble assignment that is close to the local optimal choice (by a factor of $(1+\tau)$) while also trying to keep bubble diameter and excess load as low as possible in order to ensure the bubbles are flexible enough for future patient arrivals.

Pseudocode for PairSelection-Random and PairSelection-Greedy are provided in the appendix, whereas PairSelection-τ-Greedy is shown below (Algorithm 2).

$$\min \quad \sum_{v,v' \in \mathcal{V}} (\mathbf{d}_{\mathbf{v}}^{\mathbf{ns}} \cdot \mathbf{d}_{\mathbf{v}'}^{\mathbf{ns}}) \cdot T(I_v, I_{v'}) \cdot e_{v,v'} \tag{4}$$

$$\text{s.t.} \quad u_{v,\ell} + u_{v',\ell} \leq 1 \qquad \forall \ell \in \mathcal{R}, \forall v, v' \in \mathcal{V} : I_v \cap I_{v'} \neq \emptyset \tag{5}$$

$$\sum_{k \in [K]} y_{v,k} = 1 \qquad \forall v \in \mathcal{V} \tag{6}$$

$$e_{v,v'} \geq y_{v,k} - y_{v',k} \text{ and } e_{v,v'} \geq y_{v',k} - y_{v,k} \qquad \forall k \in [K], \forall v, v' \in \mathcal{V} : I_v \cap I_{v'} \neq \emptyset \tag{7}$$

$$(u_{v,\ell} + y_{v,k} + u_{v',\ell'} + y_{v',k} - 3) \cdot D(\ell, \ell') \leq D^* \qquad \forall \ell, \ell' \in \mathcal{R}, \forall k \in [K], \forall v, v' \in \mathcal{V} : I_v \cap I_{v'} \neq \emptyset \tag{8}$$

$$\sum_{v \in \mathcal{V}} \|\mathbf{d}_v^{\mathbf{s}}\|_1 \cdot y_{v,k} - \sum_{p \in s(B^k)} L_p \leq L^* \qquad \forall k \in [K] \tag{9}$$

Algorithm 2 PairSelection-τ-Greedy($\mathcal{M}$, $(\mathcal{B}_j)_{j=1}^{i-1}$, τ)

1: $\ell^*, k^* \leftarrow \underset{\ell,k \in \mathcal{F}}{\text{argmin}}\, f((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$
2: $\mathcal{F}' \leftarrow \{(\ell, k) \mid f((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k) \leq (1+\tau) \cdot f((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)\}$
3: $\ell', k' \leftarrow \underset{\ell,k \in \mathcal{F}'}{\text{argmin}}\, C((\mathcal{B}_j)_{j=1}^{i-1}, \ell, k)$
4: **return** ℓ', k'

Optimal Offline Solution

In this section we define an integer linear program (ILP) that models the *offline* version of the bubble clustering problem, in which the entire event sequence is known ahead of time. Our hardness results (Theorems 1, 2) have already shown that in the worst case, *any* online algorithm can perform extremely poorly relative to the offline optimum. Being able to obtain an optimal offline solution via the ILP described here enables an empirical comparison over real-world input sequences, which may be structurally quite different from worst case instances. Note that even the offline bubble clustering problem is computationally hard, i.e., NP-complete [12], and thus the ILP provided here can only be solved exactly for instances of moderate size. Our ILP uses the following 3 types of 0-1 decision variables:

- $u_{v,\ell} = 1$ if and only if visit v is assigned to location ℓ.
- $y_{v,k} = 1$ if and only if visit v is assigned to bubble k.
- $e_{v,v'} = 1$ if and only if visits v, v' are assigned to different bubbles; $e_{v,v'}$ is only defined when v and v' overlap (*i.e.*, $I_v \cap I'_v \neq \emptyset$).

The objective function of the ILP is just the cross-bubble demand. Constraint set (5) ensures that each location hosts at most one visit at a time and constraint set (6) ensures that each visit is assigned to exactly one bubble. Constraint set (7) forces consistency among the e-variables and y-variables – specifically, $e_{v,v'} = 1$ iff the y-variables indicate that visits v and v' are assigned to different bubbles. Constraint set (8) ensures that the diameter upper bound D^* is satisfied. Specifically, the expression $u_{v,\ell} + y_{v,k} + u_{v',\ell'} + y_{v',k}$ in this constraint equals 4 if locations ℓ and ℓ' are assigned the same bubble k; otherwise, this expression is at most 3. Finally, the constraint set (9) ensures that total excess load for each bubble is bounded above by L^*. In this expression, we note that HCPs are pre-assigned to bubbles, and $s(B^k)$ refers to the substitutable HCPs assigned to bubble B^k.

RESULTS

Agent-Based Disease Model: We evaluate the online bubble clustering method by simulating COVID-19 spread in a MICU unit. We employ a similar model to that of [19], where each infected individual experiences a non-infectious latent period η, after which their infectivity increases exponentially until symptoms start, then decreases exponentially. After the infectivity period, each infected individual recovers with perfect immunity. This aspect of the model reflects the time-varying viral load observed in COVID-19 [5, 13, 25, 30, 35]. We denote a fixed parameter ψ as the baseline infectivity, and time-since-infection dependent infectivity scaling parameter as β. The mobility log encodes a series of interactions between individuals, each of which holds the potential for transmitting infection. Specifically, for each interaction $(v, v') \in E$ with corresponding duration $\varphi(e)$ for each infected individual v, where one individual v incident on e is infected and the other individual v' is susceptible, the probability of v successfully infecting v' during interaction (v, v') is $1 - (1 - \psi \cdot \beta)^{\varphi(e)}$.

We refer to a single simulation run as a *replicate*, which starts with a single infected substitutable HCP selected uniformly at random. Given the stochastic nature of the simulations, we run 2,500 replicates for each setting, where each replicate is a single iteration of the disease simulation. Seed values for each replicate are set using the simulation parameters (included in the Code Appendix[1]).

Data: Here, we demonstrate the performance and behavior of the proposed algorithms in (anonymized) real world hospital operation and mobility data collected from a Medical Intensive Care Unit (MICU) in a large tertiary care hospital. The HCP mobility data was collected by deploying wireless sensor devices to monitor the location of HCPs for 10 consecutive days. The *beacons* (static sensors) were placed in various locations within the unit (including patient rooms) and the *badges* (mobile sensors) were worn by HCPs. Badges transmitted probes to beacons and other badges at regular intervals, which provides fine-grained location of each HCP. Our data consists of one day of HCP mobility data repeated for 30 consecutive days; the repetition is necessary as the badges were distributed randomly among HCPs each day to ensure privacy. The patient events were separately sampled from anonymized admission-discharge-transfer (ADT) records for 30 consecutive days from the same MICU. The overlay of these two datasets provides us with a realistic view of 30-day period of interactions between 25 Nurses, 12 Specialists, and

[1] https://zenodo.org/records/18101989

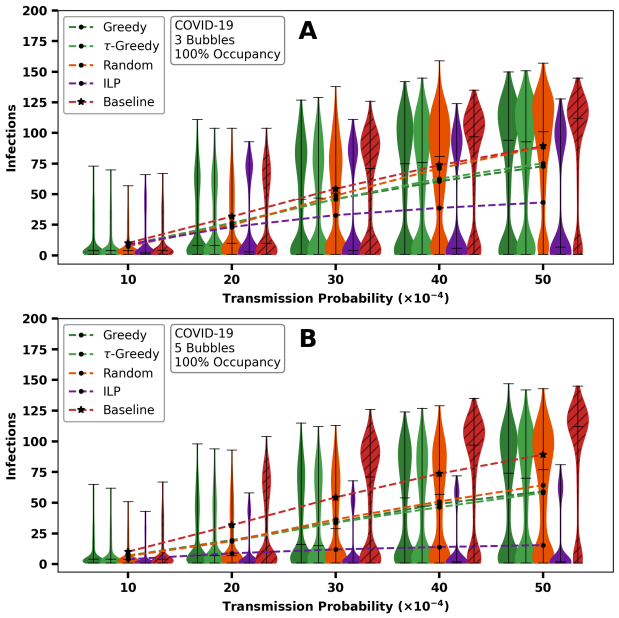

Figure 2: (Best viewed in color) Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ for (A) 3 bubbles and (B) 5 bubbles.

20 patient rooms which were dynamically occupied (and vacated) by 201 unique patient visits. We also extracted the number of HCPs of all types, their loads, and patient demands from this data, which are input to the online algorithms.

Experimental Setup: We used an agent-based COVID-19 simulation [19] on the data described above to compare the performance (w.r.t to infection control) of the proposed approaches. The details of the COVID-19 model is in the Technical Appendix. All of our experiments involve simulating the COVID-19 outbreaks (with varying parameters) in the Baseline (unmodified) mobility log and alternate mobility logs produced by each online algorithm. For smaller problem instances, we also run the ILP method, which has full access to patients' arrival and discharge times, as well as their care demands. It serves as a benchmark for the best possible outcome. While we cannot share our data due to privacy concerns, we release our code publicly in the Code and Data Appendix and share detailed experimental setup in the Technical Appendix.

Online Bubble Clustering Reduces Infection

We first evaluate the impact of online bubble clustering on the prevalence of COVID-19 infections within the MICU operating at 100% occupancy. First, we set the number of bubbles to $K = 3$ (also repeated for $K = 5$). Patients are clustered using Random, Greedy, and τ-Greedy, which are benchmarked against the Baseline (no clustering) and the offline ILP approach. The resulting clusters are then assessed using COVID-19 simulations. For these simulations, we assume perfect knowledge of each incoming patient's care needs, referred to as their demand. We evaluate the infection prevalence outcomes for COVID-19 transmissibility values of $\beta \in [1, 2, 3, 4, 5] \cdot 10^{-3}$. Results for other MICU occupancy levels and numbers of bubbles are provided in the Technical Appendix.

Each violin plot in Fig 2 represents the distribution of infection prevalence for a given setting, defined by transmissibility, number of bubbles, and MICU occupancy. Overlaid line plots indicate the average infection prevalence for each method across these settings. Notably, the violin plots in Fig 2 often display bulges at two distinct levels of prevalence, reflecting common outcomes of COVID-19 transmission. At low transmissibility ($\beta = 10^{-3}$), the prevalence distributions resulting from the Baseline, ILP, and online bubble clustering methods are nearly identical, as shown in Fig 2A and 2B. However, as β increases, the average prevalence from bubble clustering methods begins to diverge from the Baseline. For the highest transmissibility value considered ($5 \cdot 10^{-3}$), bubble clustering methods significantly reduce average COVID-19 prevalence. Using 5 bubbles (2B), clustering reduces infections from 80 to 50 on average, while using 3 bubbles (2A) reduces infections from 80 to 70. We evaluate the effect of K on infections using Welch's two-sample t-test [32, 33]. Most comparisons show significant differences in mean infections ($p < 0.05$) across K; exceptions occur at low infectivity for 5 and 7 bubbles, where infection counts are near zero. All p-values are reported in the Supplementary Information.

The effectiveness of online bubble clustering is further highlighted by the narrower bulges in the prevalence distributions for the clustering methods compared to the Baseline (shown in red). These narrower distributions indicate that high infection prevalence outcomes are much less common when clustering is applied. These results suggest that clustering patients into bubbles as they arrive can significantly lower the risk of widespread infections in a MICU, particularly under high transmissibility scenarios. The similar outcomes achieved by the three heuristics imply that clustering plays a key role, whereas the choice of specific strategy has a more limited influence on infection control.

Online Bubble Clustering Remains Feasible under Uncertain Future Patient Demand

So far, we have evaluated outcomes assuming perfect knowledge of the care required by each incoming patient. However, in real-world applications, complete information like this is rarely available. To account for this limitation, we introduce noise into the demand information of incoming patients, simulating the uncertainty inherent in such scenarios.

Let $\mathbf{D}$ represent the demand matrix, where D_{ij} denotes the demand that patient i places on non-substitutable staff member j. For each arriving patient with demand D_{ij}, we perturb the demand by sampling from a multivariate normal distribution, $\mathcal{N}(\mu, \rho\Sigma)$. Here, μ_j is the mean demand on non-substitutable staff member j, $\Sigma = \text{cov}(\mathbf{D})$ represents the covariance matrix of the demand data, and ρ is the perturbation scaler, which controls the level of noise added to the demand. We note that about 70% of covariance matrix elements have absolute value less than 2.

We then reevaluate the performance of the online bubble clustering methods under these noisy patient demands. The evaluation is conducted over 5 bubbles at a transmissibility level of $\beta = 0.004$ and MICU occupancy levels of 100%, 90%, 80%, and 70%.

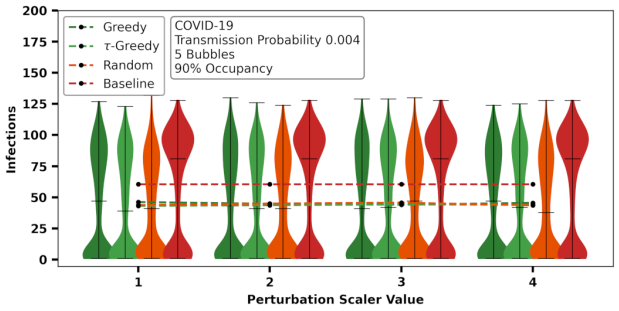

Figure 3: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta = 0.004$ over 90% MICU occupancy split into 5 bubbles

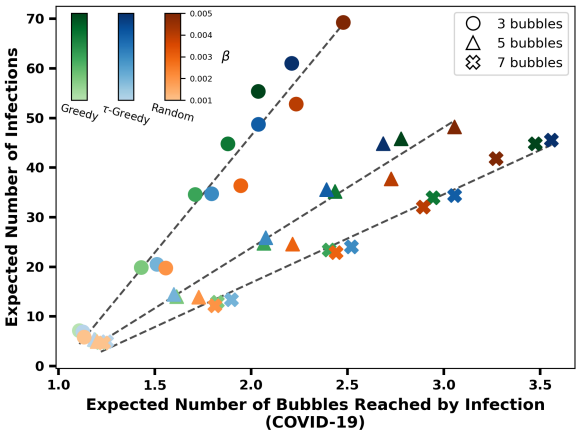

Figure 4: Expected number of bubbles reached by infection (over 2500 replicates) vs. expected number of infections.

The results of these evaluations are presented in Fig 3. Each point on the x-axis represents an increasing perturbation scaler ρ, which reflects the level of uncertainty in patient demand. Despite the increasing perturbation, we observe no corresponding rise in the expected number of infections at any occupancy level. For comparison, we plot the Baseline prevalence, which remains constant regardless of ρ as a horizontal reference line along the x-axis. This line indicates the average infection prevalence under the Baseline. Additionally, as MICU occupancy increases, the gap between the average prevalence for the online bubble clustering methods and the Baseline widens. These trends suggest that online bubble clustering remains effective even under increasingly uncertain demand information for arriving patients.

Total Infections are Highly Correlated with the Number of Bubbles the Infection Reaches

Here, we provide additional evidence for the relationship between online bubble clustering and the total expected number of infections. To this end, we compare the number of bubbles reached by infection and the total expected number of infections. Fig 4 shows the average number of bubbles affected by COVID-19 (from a single source infection) versus the expected number of infections for each setting.

Fig 4 illustrates that as transmissibility increases (indicated by the darker shades of each point), there is a strong correlation between the expected number of COVID-19 infections and the number of bubbles reached by the infection. Notably, even at the highest transmissibility levels, infection is not expected to reach all bubbles, regardless of the number of bubbles into which arriving patients are divided. High Pearson correlation coefficient values (included in the Technical Appendix) further demonstrate that the overall prevalence of infection is strongly influenced by the number of bubbles used for clustering.

Cost Analysis Across Settings and Methods

Compartmentalizing hospital staff inevitably reduces flexibility in patient care. Here, we focus on the cost of clustering patients and staff, measured in terms of HCP load. Specifically, we compare the mean Load for HCPs for $k = 5$ at 90% occupancy, as shown in Table 1. Among the methods, Random achieves the lowest total HCP load but results in a slightly higher expected number of infections

Table 1: Expected number of infections and total HCP load.

K=5 90% Occupancy	# of Infections Mean ± STD	Load Mean ± STD
Baseline	74.7 ± 45.4	1444.7 ± 48.5
Greedy	45.8 ± 38.6	1184.7 ± 133.6
τ-Greedy	44.8 ± 39.7	1193.3 ± 134.9
Random	48.2 ± 37.4	1177.7 ± 154.1

compared to Greedy and τ-Greedy (by only 3 for COVID-19). Interestingly, we observe that τ-Greedy does not result in a lower HCP load compared to Greedy, despite its design to balance infection reduction with cost minimization. This outcome is accompanied by a fair amount of variability in the simulation results. Despite not being our primary objective, it is worth noting that these methods also result in a lower HCP load compared to the Baseline.

CONCLUSION AND DISCUSSION

In this paper, we demonstrate the effectiveness of simple real-time bubble-clustering methods to limit COVID-19 spread among patients and hospital staff. We establish that the hardness of the online bubble clustering problem, compared to the offline version, is dependent on circumstances and context. However, we also empirically demonstrate that simple approaches, such as greedy and random methods, can significantly reduce the burden of COVID-19 spread within a MICU. Wherever viable, we additionally compute optimal offline solutions for comparison with the online bubble clustering methods to show the inherent difficulty of real-time clustering.

Previous research in this area assumed an offline setting in bubble clustering [12], whereas this work seeks to reflect the realities of real-time patient assignments. Our experiments have found that online clustering methods of assigning patients and HCPs serve to decrease the prevalence of disease in a MICUs, while still maintaining a robust response to the uncertainties of care required by

incoming patients. Remarkably, the performance of random clustering methods in our experiments were on par with that of greedy methods, suggesting that even naive approaches are effective, and that clustering care in the first place is the most important part. The real-time clustering methods' simplicity can potentially help with their implementation and integration in existing staff and patient assignment systems.

Since it is generally infeasible to identify and isolate infections as they occur, we develop methods for strategically placing staff and incoming patients in advance to reduce transmission throughout the unit. The results should be interpreted with a few limitations in mind. First, we only tested our methods for the spread of COVID-19. In future work, the range of studied diseases could be expanded to include hospital-acquired infections, such as Clostridioides difficile, Vancomycin-resistant Enterococcus, or Methicillin-resistant Staphylococcus [1, 3, 14, 15, 27]. However, broadening the scope of study may result in significantly different outcomes, as infections that spread more easily than COVID-19 via fomites would likely impact the effectiveness of clustering methods alone without additional assumptions regarding routine cleaning. Our project was also limited to mobility and contact data from a single MICU; as such, establishing the portability of our methods is an important step towards a framework for on-the-ground implementation. The primary difficulty in applying these methods to any unit is establishing the existing contact patterns of patients and staff within that unit. We also assume that only information about the next patient arrival is available at any given time. In practice, hospitals may have advance knowledge of pending patients or can assign patients based on historical demand patterns. While this assumption simplifies our model, it could be relaxed in future work to allow access to information about multiple upcoming arrivals or historical demand patterns to potentially improve performance. Finally, this project did not consider costs beyond those based on the staff load and distance between rooms. Future exploration could include the incorporation of more sophisticated cost measures, such as any unmet demand or decrease in quality of care experienced by patients.

ACKNOWLEDGMENTS

The authors acknowledge feedback from members of the Computational Epidemiology research group at the University of Iowa and the CDC MInD-Healthcare group.

REFERENCES

[1] Cybele L Abad, Anna K Barker, and Nasia Safdar. 2020. A systematic review of the effectiveness of cohorting to reduce transmission of healthcare-associated C. difficile and multidrug-resistant organisms. *Infection Control & Hospital Epidemiology* 41, 6 (2020), 691–709.

[2] Bijaya Adhikari, Bryan Lewis, Anil Vullikanti, José Mauricio Jiménez, and B Aditya Prakash. 2019. Fast and near-optimal monitoring for healthcare acquired infection outbreaks. *PLoS computational biology* 15, 9 (2019), e1007284.

[3] D. J. Austin, M. J. M. Bonten, R. A. Weinstein, S. Slaughter, and R. M. Anderson. 1999. Vancomycin-resistant enterococci in intensive-care hospital settings: Transmission dynamics, persistence, and the impact of infection control programs. *Proceedings of the National Academy of Sciences* 96, 12 (1999), 6908–6913. https://doi.org/10.1073/pnas.96.12.6908

[4] Aditya Bhaskara and Aravinda Kanchana Ruwanpathirana. 2020. Robust Algorithms for Online k-means Clustering. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory (Proceedings of Machine Learning Research, Vol. 117)*, Aryeh Kontorovich and Gergely Neu (Eds.). PMLR, 148–173. https://proceedings.mlr.press/v117/bhaskara20a.html

[5] M. Cevik, M. Tate, O. Lloyd, A. E. Maraolo, J. Schafers, and A. Ho. 2021. SARS-CoV-2, SARS-CoV, and MERS-CoV viral load dynamics, duration of viral shedding, and infectiousness: a systematic review and meta-analysis. *The Lancet Microbe* 2, 1 (Jan 2021), e13–22. https://doi.org/10.1016/S2666-5247(20)30172-5

[6] Joseph D Forrester, Paul M Maggio, and Lakshika Tennakoon. 2022. Cost of health care–associated infections in the United States. *Journal of patient safety* 18, 2 (2022), e477–e479.

[7] Alex W Friedrich. 2019. Control of hospital acquired infections and antimicrobial resistance in Europe: the way to go. *Wiener Medizinische Wochenschrift* 169, Suppl 1 (2019), 25–30.

[8] William Gardner, David States, and Nicholas Bagley. 2020. The coronavirus and the risks to the elderly in long-term care. *Journal of aging & social policy* 32, 4-5 (2020), 310–315.

[9] Gurobi Optimization, LLC. 2024. *Gurobi Optimizer Reference Manual.* https://www.gurobi.com.

[10] Morcos Hanna, Rita Shah, Lucila Marquez, Rebecca Barzegar, Adrienne Gordon, and Mohan Pammi. 2023. Infant isolation and cohorting for preventing or reducing transmission of healthcare-associated infections in neonatal units. *Cochrane Database of Systematic Reviews* 6 (2023).

[11] DM Hasibul Hasan, Alex Rohwer, Hankyu Jang, Ted Herman, Philip M Polgreen, Daniel K Sewell, Bijaya Adhikari, and Sriram V Pemmaraju. 2021. Modeling and evaluation of clustering patient care into bubbles. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*. IEEE, 73–82.

[12] D. M. Hasibul Hasan, A. Alex Rohwer, H. Jang, T. Herman, P. M. Polgreen, D. K. Sewell, B Adhikari, and S. V. Pemmaraju. 2021. Modeling and Evaluation of Clustering Patient Care into Bubbles. https://doi.org/10.1109/ICHI52183.2021.00023. In *AIEEE 9th International Conference on Healthcare Informatics*. ICHI, 73–82.

[13] D. He, S. Zhao, Q. Lin, Z. Zhuang, P. Cao, M. H. Wang, et al. 2020. The relative transmissibility of asymptomatic COVID-19 infections among close contacts. *International Journal of Infectious Diseases* 94 (2020), 145–147. https://doi.org/10.1016/j.ijid.2020.04.034

[14] H. Humphreys, H. Grundmann, R. Skov, J.-C. Lucet, and R. Cauda. 2009. Prevention and control of methicillin-resistant Staphylococcus aureus. *Clinical Microbiology and Infection* 15, 2 (2009), 120–124. https://doi.org/10.1111/j.1469-0691.2009.02699.x

[15] K. Hussein, G. Rabino, O. Eluk, S. Warman, S. Reisner, Y. Geffen, L. Halif, and M. Paul. 2017. The association between infection control interventions and carbapenem-resistant Enterobacteriaceae incidence in an endemic hospital. *Journal of Hospital Infection* 97, 3 (2017), 218–225. https://doi.org/10.1016/j.jhin.2017.07.018

[16] J Islam, E Cheek, V Navani, C Rajkumar, J Cohen, and MJ Llewelyn. 2013. Influence of cohorting patients with Clostridium difficile infection on risk of symptomatic recurrence. *Journal of Hospital Infection* 85, 1 (2013), 17–21.

[17] Hankyu Jang, Andrew Fu, Jiaming Cui, Methun Kamruzzaman, B Aditya Prakash, Anil Vullikanti, Bijaya Adhikari, and Sriram V Pemmaraju. 2023. Detecting sources of healthcare associated infections. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4347–4355.

[18] Hankyu Jang, Shreyas Pai, Bijaya Adhikari, and Sriram V Pemmaraju. 2021. Risk-aware temporal cascade reconstruction to detect asymptomatic cases: For the cdc mind healthcare network. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 240–249.

[19] H. Jang, P. M. Polgreen, A. M. Segre, and S. V. Pemmaraju. 2021. COVID-19 modeling and non-pharmaceutical interventions in an outpatient dialysis unit. *PLOS Computational Biology* 17, 7 (2021), e1009177. https://doi.org/10.1371/journal.pcbi.1009177

[20] José Mauricio Jiménez. 2014. *The utilization of macroergonomics and simulation to improve control of healthcare acquired infections.* Ph.D. Dissertation. Virginia Polytechnic Institute and State University.

[21] David S. Johnson, Alan Demers, Jeffrey D. Ullman, Michael R Garey, and Ronald L. Graham. 1974. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on computing* 3, 4 (1974), 299–325.

[22] Minji Kang, Madhuri B Nagaraj, Krystle K Campbell, Ian A Nazareno, Daniel J Scott, Doramarie Arocha, and Julie B Trivedi. 2022. The role of simulation-based training in healthcare-associated infection (HAI) prevention. *Antimicrobial Stewardship & Healthcare Epidemiology* 2, 1 (2022), e20.

[23] Abhijeet Kharkar, D M Hasibul Hasan, Philip Polgreen, Alberto Segre, Daniel Sewell, and Sriram Pemmaraju. 2020. Naturally Emerging Cohorting Behavior of Healthcare Workers and Its Implications for Disease Spread. *Infection Control & Hospital Epidemiology* 41, S1 (2020), s329–s330. https://doi.org/10.1017/ice.2020.932

[24] Timothy Lawes, José-María Lopez-Lozano, Cesar A Nebot, Gillian Macartney, Rashmi Subbarao-Sharma, Ceri RJ Dare, Karen D Wares, and Ian M Gould. 2015. Effects of national antibiotic stewardship and infection control strategies on hospital-associated and community-associated meticillin-resistant Staphylococcus aureus infections across a region of Scotland: a non-linear time-series study. *The Lancet Infectious Diseases* 15, 12 (2015), 1438–1449.

[25] F.-X. Lescure, L. Bouadma, D. Nguyen, M. Parisey, P.-H. Wicky, S. Behillil, et al. 2020. Clinical and virological data of the first cases of COVID-19 in Europe: a case series. *Lancet Infectious Diseases* 20, 6 (Mar 2020), 697–706. https://doi.org/10.1016/S1473-3099(20)30200-0
[26] V Mouajou, K Adams, G DeLisle, and C Quach. 2022. Hand hygiene compliance in the prevention of hospital-acquired infections: a systematic review. *Journal of Hospital Infection* 119 (2022), 33–48.
[27] J. Price, E. Cheek, S. Lippett, M. Cubbon, D. N. Gerding, S. P. Sambol, D. M. Citron, and M. Llewelyn. 2010. Impact of an intervention to control *Clostridium difficile* infection on hospital- and community-onset disease; an interrupted time series analysis. *Clinical Microbiology and Infection* 16, 8 (2010), 1297–1302. https://doi.org/10.1111/j.1469-0691.2009.03077.x
[28] Samira Raoofi, Fatemeh Pashazadeh Kan, Sima Rafiei, Zahra Hosseinipalangi, Zahra Noorani Mejareh, Saghar Khani, Bahare Abdollahi, Fatemeh Seyghalani Talab, Mohaddeseh Sanaei, Farnaz Zarabi, et al. 2023. Global prevalence of nosocomial infection: A systematic review and meta-analysis. *PloS one* 18, 1 (2023), e0274248.
[29] R Douglas Scott. 2009. The direct medical costs of healthcare-associated infections in US hospitals and the benefits of prevention. (2009).
[30] K. K.-W. To, O. T.-Y. Tsang, W.-S. Leung, A. R. Tam, T.-C. Wu, D. C. Lung, et al. 2020. Temporal profiles of viral load in posterior oropharyngeal saliva samples and serum antibody responses during infection by SARS-CoV-2: an observational cohort study. *Lancet Infectious Diseases* 20, 5 (May 2020), 565–. https://doi.org/10.1016/S1473-3099(20)30196-1
[31] Rahul Vaze. 2023. *Online Algorithms.* Cambridge University Press.
[32] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17, 3 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2
[33] Bernard L. Welch. 1947. The Generalization of "Student's" Problem when Several Different Population Variances are Involved. *Biometrika* 34, 1/2 (1947), 28–35. https://doi.org/10.1093/biomet/34.1-2.28
[34] Andrew C. Yao. 1977. Probabilistic Computations: Toward a Unified Measure of Complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE, 222–227. https://doi.org/10.1109/SFCS.1977.24
[35] L. Zou, F. Ruan, M. Huang, L. Liang, H. Huang, Z. Hong, et al. 2020. SARS-CoV-2 Viral Load in Upper Respiratory Specimens of Infected Patients. *New England Journal of Medicine* 382, 12 (Mar 2020), 1177–1179. https://doi.org/10.1056/NEJMc2001737

A ALGORITHMIC APPROACH

We provide pseudocode for a more formal "Online Processing of Patient Events" algorithm in 3, as well as the RANDOM and GREEDY subroutines for PAIRSELECTION in algorithms 4 and 5, respectively.

Algorithm 3 BUBBLECLUSTER($\mathcal{M}$, PAIRSELECTION($\cdot$))

$\mathcal{B} \leftarrow \{\{\emptyset \mid k \in [K]\} \mid t \in [T]\}$
for $t \in [T]$ **do**
  if $\mathcal{E}_t$ is a discharge of v from ℓ **then**
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{\ell\}$
    $\mathcal{B}^t(\ell) \leftarrow \mathcal{B}^t(\ell) \setminus \ell$
  else
    $\ell', k' \leftarrow$PAIRSELECTION$(\mathcal{M}, \mathcal{B}^t, \tau, \alpha)$
    $\mathcal{B}^t_{k'} \leftarrow \mathcal{B}^t_{k'} \cup \{\ell'\}$
    $\mathcal{U} \leftarrow \mathcal{U} \backslash \ell'$
  end if
end for

Algorithm 4 PAIRSELECTION-RANDOM($\mathcal{M}, \mathcal{B}^t$)

$\mathcal{F} \leftarrow \{(\ell, k) \mid D(\mathcal{B}^t_k \cup \{\ell\}) \leq D^*, \hat{L}(\mathcal{B}^t_k \cup \{\ell\}) \leq L^*\}$
$\ell^*, k^* \sim \text{Uniform}(\mathcal{F})$
return ℓ^*, k^*

Algorithm 5 PAIRSELECTION-GREEDY($\mathcal{M}, \mathcal{B}^t, \tau, \alpha$)

$\mathcal{F} \leftarrow \{(\ell, k) \mid D(\mathcal{B}^t_k \cup \{\ell\}) \leq D^*, \hat{L}(\mathcal{B}^t_k \cup \{\ell\}) \leq L^*\}$
$\ell^*, k^* \leftarrow \underset{\ell, k \in \mathcal{F}}{\text{argmin}}\ f^t(\mathcal{B}^t_k \cup \{\ell\})$
return ℓ^*, k^*

RESULTS

Mobility Log: Our methods are highly dependent on capturing and modifying the movement of HCPs, as they are the primary mechanism of inter-bubble disease spread. This movement is captured by a *mobility log* that consists of information specifying a healthcare professional visits a patient room and for how long, which was obtained using a sensor system that we deployed. The data was collected over 30 days at the MICU in a large tertiary care hospital in the U.S. Midwest and has 6202 records for 265 patients containing information on patient room assignment, nurse assignment, admission, and discharge. Interactions between hospital staff in non-patient room locations (e.g., hallways, break-rooms, and nurses stations) are also present. The mobility log is represented by an edge-labeled bipartite multigraph $G = (P, \mathcal{R}, E, \Phi)$, which we refer to as a *visit graph*, similar to the one described in [12]. Suppose P denotes the set of HCPs, $\mathcal{R}$ is the set of locations, and E is a set of edges specifying HCP visits to rooms. Each edge $e = \{p, \ell\}$ ($p \in P, \ell \in \mathcal{R}$) has a label $\varphi(e) = (start(e), end(e))$ to specify to start and end time of the visit. We assume that each HCP cannot be in multiple locations simultaneously, that is, $(start(e), end(e)) \cap (start(e'), end(e')) = \emptyset$ for any two edges e and e' incident on an HCP p.

Reproducibility

Experimental Systems: All experiments were run on AMD EPYC 7763 CPUs with 2 TB RAM. All code was implemented in Python, and non-standard Python libraries used include `matplotlib`, `seaborn`, `pickle`, `networkx`, `numpy`, `pandas`, and `scipy`. The offline ILP was implemented and executed in `Gurobi` [9].
Code and Data Availability: Our data processing and experimental code is available as part of the Supplementary Information. Due to the sensitive nature of our collected data, we cannot release the original dataset to protect HCP and patient privacy as per our IRB. We include a toy dataset that exactly matches the format of the full dataset in the Code Appendix.
Parameters: We specify the parameters used for each experiment in Table 2.

Online Bubble Clustering Reduces Infection

Here we provide additional experimental results for a wider range of transmissibilities, occupancies, and number of bubbles. Results for 3 bubbles can be found in Fig 5, 5 bubbles in Fig 6, and 7 bubbles in Fig 7. In addition, Tables 3 - 6 provide Welch's two-sample t-test p-values for the difference in infection means between K values (number of bubbles). Statistically significant ($p < 0.05$) values in each table are bold.

Online Bubble Clustering Remains Feasible under Uncertain Future Patient Demand

Full experimental results for 3 bubbles can be found in Fig 8, 5 bubbles in Fig 9, and 7 bubbles in Fig 10.

Total Infections are Highly Correlated with the Number of Bubbles the Infection Reaches

Here we provide experiments for varying occupancy. Results for 100% occupancy can be found in Fig 11, 80% occupancy in Fig 12, and 70% occupancy in Fig 13. The Pearson correlation coefficients values associated with the correlation between number of bubbles reached and total infections may be found in Table 7.

The Effectiveness of Online Bubble Clustering Depends on the Number of Bubbles and MICU Occupancy

While online bubble clustering is generally effective at reducing infection spread, its performance varies depending on the specific setting. We assess how the effectiveness of online bubble clustering is influenced by how many bubbles substitutable staff and patients are divided into, as well as the MICU occupancy level. For this analysis, patients are clustered into 3, 5, and 7 bubbles at all transmissibilities β, across MICU occupancy levels of 70%, 80% 90%, and 100%. The resulting settings are then evaluated in terms of the average proportion of infections reduced for COVID-19, as shown in Fig 14.

Fig 14 demonstrates that clustering patients into 3 bubbles in a low MICU occupancy setting is ineffective for reducing infections. However, in high occupancy settings, online bubble clustering methods can still achieve a reduction in infections by approximately 20%. In contrast, clustering patients into 5 bubbles can reduce infection prevalence by up to 50%, while using 7 bubbles can achieve reductions of up to 70%. Across most settings, the various online bubble clustering methods perform similarly, indicating that infection prevalence can be significantly reduced as long as arriving patients are assigned to clusters. Fig 14 shows results for additional values of transmissibility β.

Table 2: Parameter values used in experimental setup.

Parameter	**Values for K = 3, 5, 7**
Transmissibility β	0.001, 0.002, 0.003, 0.004, 0.005
Perturbation scaler ρ	1, 2, 3, 4
Parameter	**Values for K = 3**
α	0.3
τ	0.6
Diameter bound D^*	10
Load bound L^*	500
Parameter	**Values for K = 5**
α	0.3
τ	0.6
Diameter bound D^*	7
Load bound L^*	300
Parameter	**Values for K = 7**
α	0
τ	2.75
Diameter bound D^*	5
Load bound L^*	200

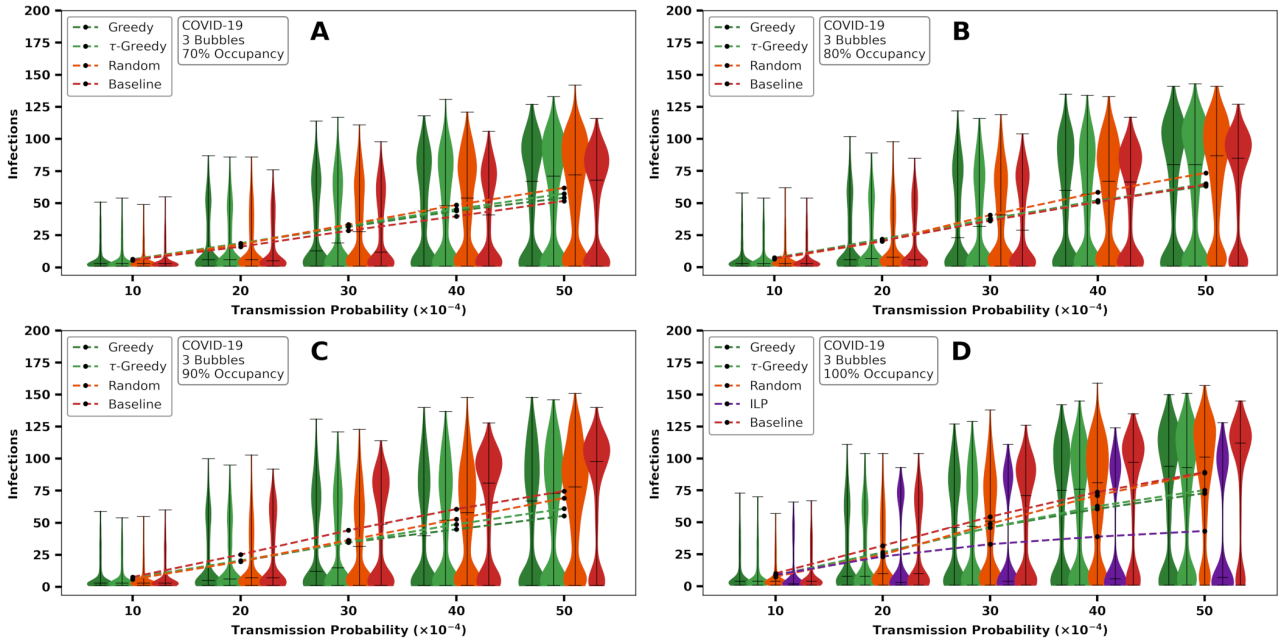

Figure 5: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ split into 3 bubbles over (A) 70%, (B) 80%, (C) 90%, and (D) 100% MICU occupancy.

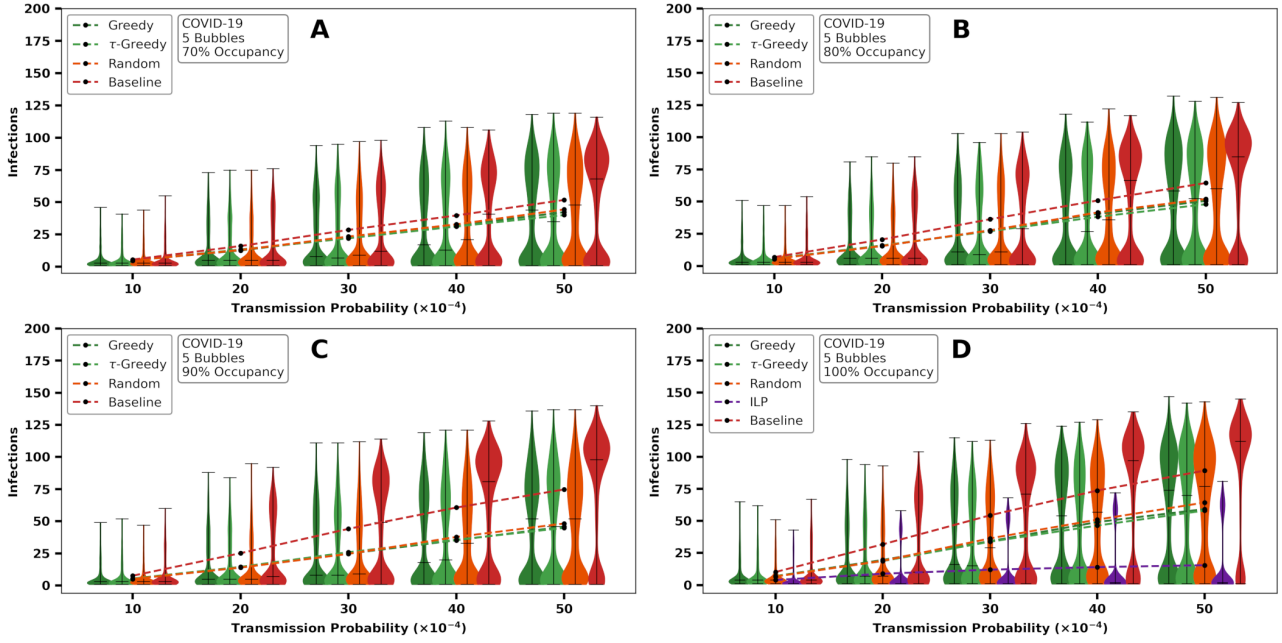

Figure 6: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ split into 5 bubbles over (A) 70%, (B) 80%, (C) 90%, and (D) 100% MICU occupancy.

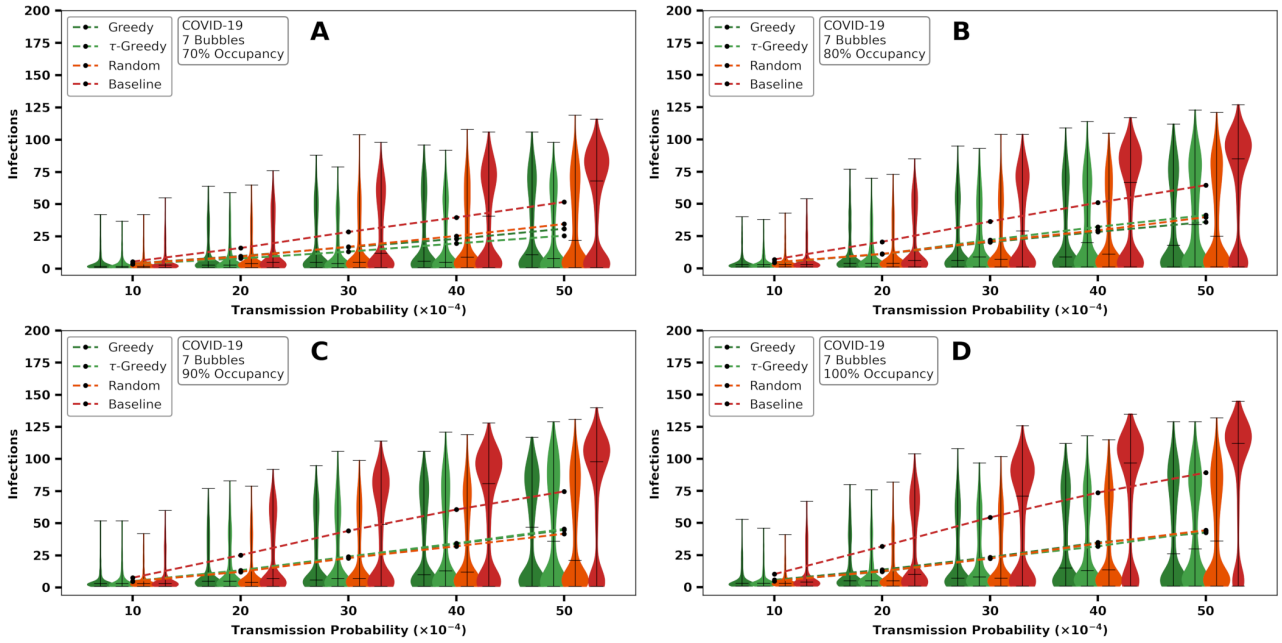

Figure 7: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ split into 7 bubbles over (A) 70%, (B) 80%, (C) 90%, and (D) 100% MICU occupancy.

Table 3: Welch's two-sample t-test p-values for the difference in infection means between K values (number of bubbles) (bold: $p < 0.05$) with 70% occupancy.

		K comparison		
β	**Method**	3 vs 5	3 vs 7	5 vs 7
0.001	Greedy	**6.73e-08**	**5.53e-10**	0.163
0.001	τ-Greedy	**0.000212**	0.349	**0.0126**
0.001	Random	**0.00361**	**0.0166**	0.783
0.002	Greedy	**1.54e-07**	**1.26e-14**	**0.00185**
0.002	τ-Greedy	**1.26e-06**	0.443	**1.26e-06**
0.002	Random	**3.11e-05**	**0.000311**	0.972
0.003	Greedy	**9.15e-24**	**1.21e-21**	0.261
0.003	τ-Greedy	**1.22e-07**	0.178	**0.00168**
0.003	Random	**2.07e-12**	**1.46e-11**	0.423
0.004	Greedy	**7.81e-28**	**6.65e-28**	0.0547
0.004	τ-Greedy	**3.43e-16**	**1.21e-05**	**0.00869**
0.004	Random	**4.19e-17**	**1.62e-29**	**3.65e-05**
0.005	Greedy	**1.06e-60**	**5.58e-64**	**4.8e-05**
0.005	τ-Greedy	**8.13e-28**	**1.17e-13**	0.0835
0.005	Random	**4.32e-49**	**4.79e-65**	**9.48e-08**

Table 4: Welch's two-sample t-test p-values for the difference in infection means between K values (number of bubbles) (bold: $p < 0.05$) with 80% occupancy.

		K comparison		
β	**Method**	3 vs 5	3 vs 7	5 vs 7
0.001	Greedy	**7.83e-14**	**3.77e-10**	0.62
0.001	τ-Greedy	**0.00014**	**5.53e-14**	**1.57e-05**
0.001	Random	**1.22e-05**	**1.67e-09**	**0.0327**
0.002	Greedy	**1.62e-12**	**1.08e-12**	0.335
0.002	τ-Greedy	**9.08e-10**	**5.7e-24**	**4.83e-07**
0.002	Random	**3.31e-08**	**4.35e-08**	0.512
0.003	Greedy	**5.89e-31**	**8.06e-37**	**0.00282**
0.003	τ-Greedy	**4.66e-10**	**1.01e-69**	**3.13e-35**
0.003	Random	**3.38e-12**	**4.08e-19**	**0.00429**
0.004	Greedy	**1.89e-54**	**1.32e-50**	**0.0101**
0.004	τ-Greedy	**1.64e-25**	**2.13e-114**	**3.97e-48**
0.004	Random	**8.39e-33**	**6.53e-31**	0.0677
0.005	Greedy	**8.46e-88**	**9.9e-82**	**0.000189**
0.005	τ-Greedy	**1.05e-40**	**1.93e-148**	**6.59e-58**
0.005	Random	**5.65e-58**	**6.97e-62**	**9.19e-05**

Table 5: Welch's two-sample t-test p-values for the difference in infection means between K values (number of bubbles) (bold: $p < 0.05$) with 90% occupancy.

		K comparison		
β	**Method**	3 vs 5	3 vs 7	5 vs 7
0.001	Greedy	**1.39e-09**	**1.81e-12**	0.122
0.001	τ-Greedy	**1.6e-05**	**1.84e-12**	**0.000763**
0.001	Random	**4.34e-06**	**1.02e-08**	0.0928
0.002	Greedy	**1.58e-12**	**6.2e-33**	**7.59e-09**
0.002	τ-Greedy	**0.000858**	**7.65e-28**	**3.81e-15**
0.002	Random	**6.42e-09**	**4.14e-13**	**0.0345**
0.003	Greedy	**2.52e-27**	**2.74e-60**	**7.3e-13**
0.003	τ-Greedy	**2.48e-17**	**3.11e-59**	**9.19e-20**
0.003	Random	**6.68e-14**	**2.28e-28**	**7.55e-06**
0.004	Greedy	**2.09e-51**	**4.07e-105**	**4.29e-21**
0.004	τ-Greedy	**6.37e-23**	**1.26e-91**	**4.18e-34**
0.004	Random	**7.2e-38**	**3.23e-42**	**0.00244**
0.005	Greedy	**5.77e-83**	**5.89e-159**	**1.65e-33**
0.005	τ-Greedy	**2.08e-36**	**1.97e-140**	**4.55e-51**
0.005	Random	**1.44e-50**	**2.94e-55**	**0.000189**

Table 6: Welch's two-sample t-test p-values for the difference in infection means between K values (number of bubbles) (bold: $p < 0.05$) with 100% occupancy.

		K comparison		
β	**Method**	3 vs 5	3 vs 7	5 vs 7
0.001	Greedy	**5.93e-12**	**1.46e-19**	**0.000926**
0.001	τ-Greedy	**4.23e-07**	**6.5e-07**	0.562
0.001	Random	**1.72e-10**	**0.000597**	**0.0171**
0.002	Greedy	**6.88e-19**	**2.3e-22**	**0.0181**
0.002	τ-Greedy	**2.53e-09**	**1.29e-11**	0.127
0.002	Random	**2.76e-13**	**8.23e-07**	0.17
0.003	Greedy	**3.17e-31**	**3.1e-35**	**0.00151**
0.003	τ-Greedy	**6.96e-19**	**3.96e-26**	**0.00226**
0.003	Random	**1.36e-36**	**3.82e-12**	**0.00109**
0.004	Greedy	**1.08e-77**	**6.74e-86**	**5.74e-08**
0.004	τ-Greedy	**1.95e-34**	**9e-49**	**4.39e-06**
0.004	Random	**1.63e-50**	**4.73e-28**	0.558
0.005	Greedy	**1.25e-94**	**8.73e-103**	**4.96e-10**
0.005	τ-Greedy	**1.36e-56**	**2.29e-98**	**2.01e-16**
0.005	Random	**7.37e-98**	**3.08e-62**	0.251

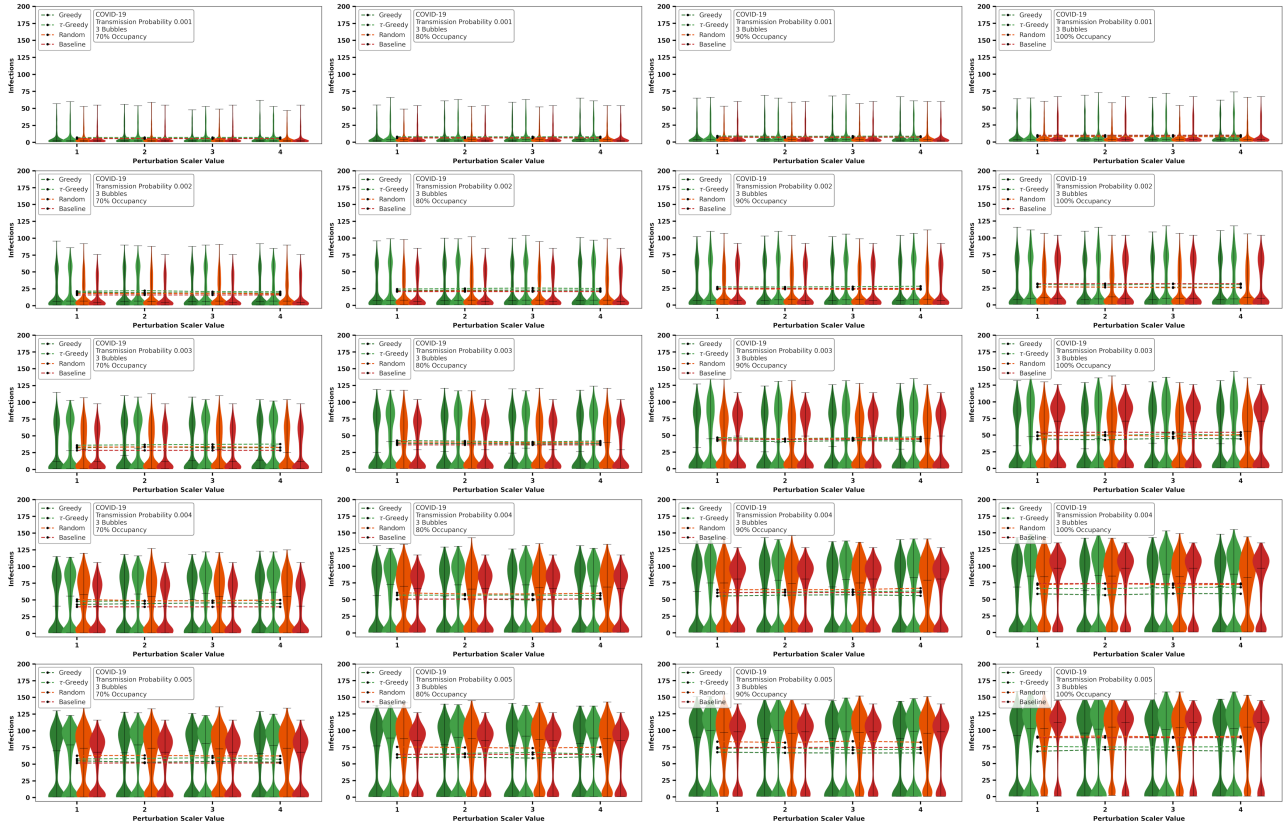

Figure 8: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta = [1, 2, 3, 4, 5] \cdot 10^{-3}$ over 70%, 80%, 90%, and 100% MICU occupancy split into 3 bubbles.

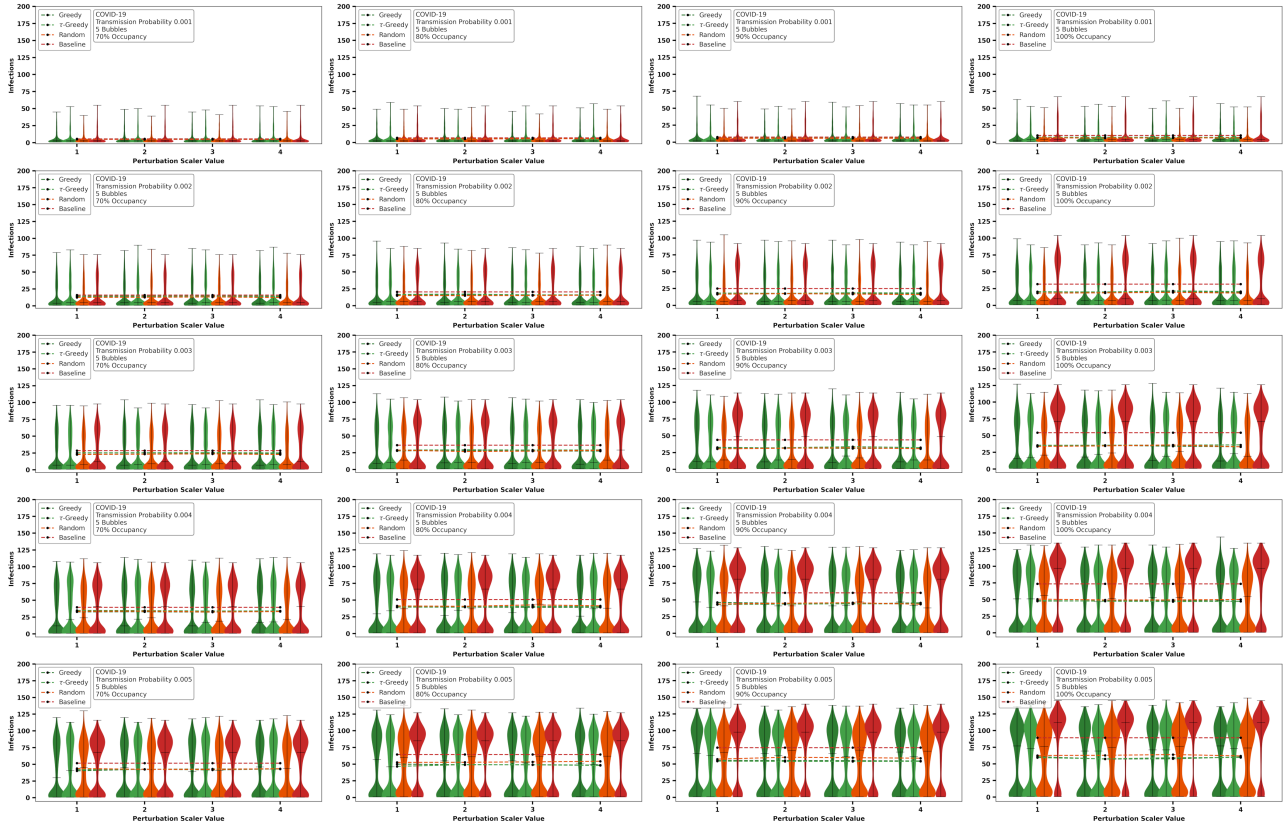

Figure 9: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta = [1, 2, 3, 4, 5] \cdot 10^{-3}$ over 70%, 80%, 90%, and 100% MICU occupancy split into 5 bubbles.

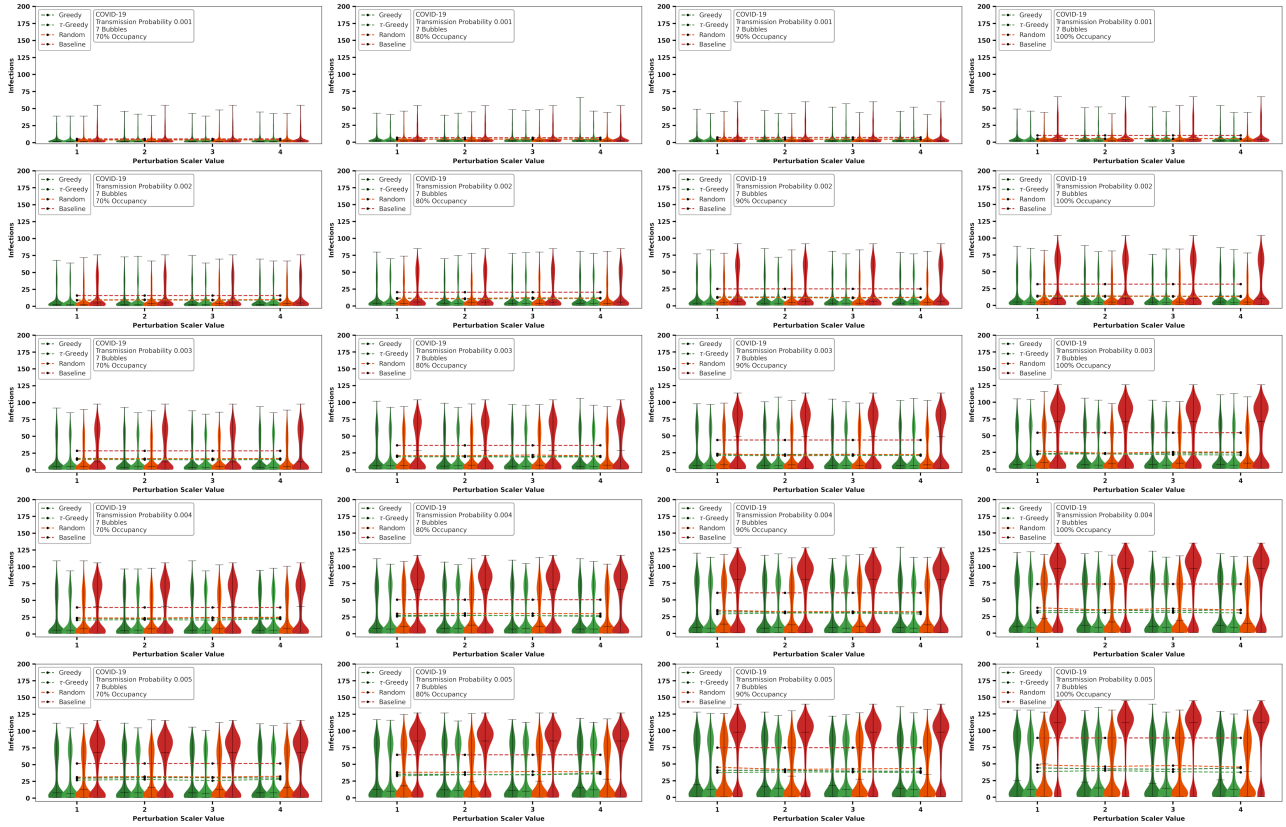

Figure 10: Ranges of infection levels (over 2500 replicates) for pathogen transmissibility $\beta = [1, 2, 3, 4, 5] \cdot 10^{-3}$ over 70%, 80%, 90%, and 100% MICU occupancy split into 7 bubbles.

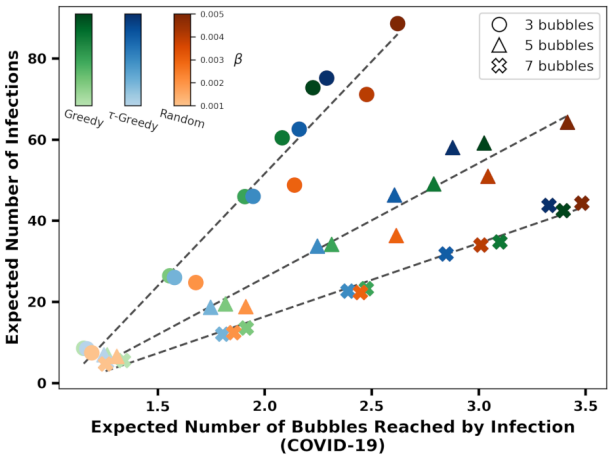

Figure 11: Expected number of bubbles reached by infection (over 2500 replicates) vs. expected number of infections for transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ over 100% MICU occupancy for COVID-19.

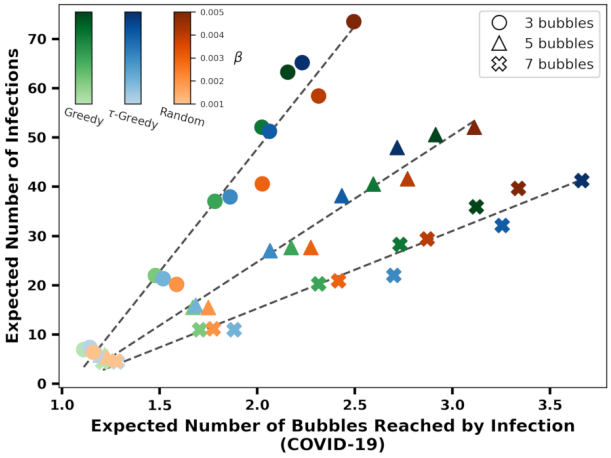

Figure 12: Expected number of bubbles reached by infection (over 2500 replicates) vs. expected number of infections for transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ over 80% MICU occupancy for COVID-19.

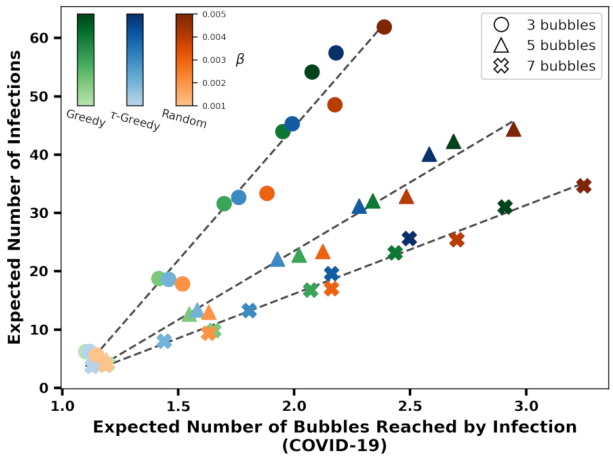

Figure 13: Expected number of bubbles reached by infection (over 2500 replicates) vs. expected number of infections for transmissibility $\beta \in [0.001, 0.002, ..., 0.005]$ over 70% MICU occupancy for COVID-19.

Table 7: Pearson Correlation Coefficient values of the number of bubbles reached by infection and number of infections for each number of bubbles × clustering method pair under 100%, 80%, and 70% occupancies.

COVID-19 100% Occupancy	Random	Greedy	τ-Greedy
$K = 3$	0.98	0.99	0.99
$K = 5$	0.99	1.0	0.99
$K = 7$	0.99	1.0	0.99
COVID-19 80% Occupancy	Random	Greedy	τ-Greedy
$K = 3$	0.99	0.99	0.99
$K = 5$	1.0	1.0	1.0
$K = 7$	0.99	1.0	0.99
COVID-19 70% Occupancy	Random	Greedy	τ-Greedy
$K = 3$	0.99	0.99	0.99
$K = 5$	1.0	1.0	1.0
$K = 7$	1.0	1.0	1.0

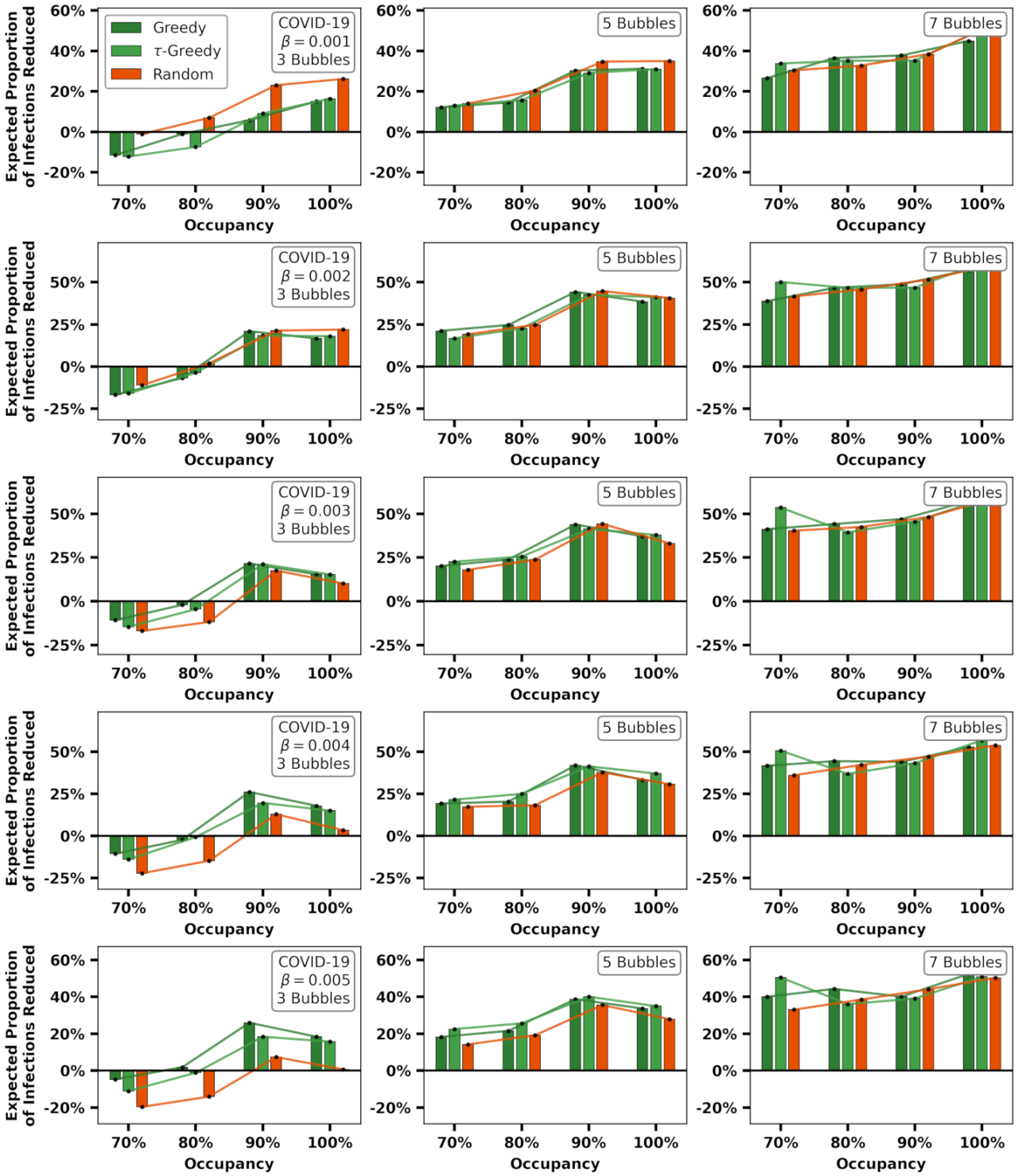

Figure 14: Methods for each occupancy level and for COVID-19.